Promises and Futures Overview Solutions

Copyright © James Raynard 2021

Producer-Consumer

- Write a multi-threaded program which uses the producer-consumer model with std::promise and std::future

Copyright © James Raynard 2020

Exceptions

- Normally, an exception handler which rethrows an exception will use either a copy of the original exception, or a reference to it
- Why does this not work with std::promise?
  - In a multi-threaded program, each thread has its own execution stack
  - std::promise is used to transfer an exception from one thread to another
  - We cannot use a reference to an exception, because the address of the exception is only valid in the original thread
  - If we use a copy of the exception, we risk "slicing" the exception. In this case, the target thread would only get the base class part of the exception

Copyright © James Raynard 2020

Producer-Consumer with Exception

- Write a multi-threaded program which uses the producer-consumer model with std::promise and std::future
- The producer thread of this program throws an exception which is handled in the consumer thread
- Check that your program runs correctly

Copyright © James Raynard 2020